M.Sc. 2nd Semester Examination, 2012
COMPUTER SCIENCE
PAPER—COS-202

Full Marks : 50

Time : 2 hours

The figures in the right-hand margin indicate marks

Candidates are required to give their answers in their own words as far as practicable

Illustrate the answers wherever necessary

(Theory of Computation)

MODULE — 1

[Marks : 25]

Answer any two questions

1. (a) Construct a deterministic finite automaton accepting the set of all strings over \{a, b\} ending with bba.
(b) Differentiate between Mealy machine and Moore machine.

(c) Construct a grammar $G$ that will generate the following language $L$ over \{a, b\}:

$$L(G) = \{a^n b^n c^i \mid n \geq 1, \ i \geq 0\}.$$  

2. (a) Prove that the language $L = \{a^p \mid P \text{ is prime}\}$ is not regular.

(b) Construct NFA equivalent to the following R.E

$$10 + (0 + 11) 0^* 1$$

3. (a) Show that the following grammar is ambiguous:

$$S \rightarrow a \mid abSb \mid aAb$$

$$A \rightarrow bS \mid aAAb$$

(b) Convert the grammar $S \rightarrow AB, \ A \rightarrow BS \mid b,$

$B \rightarrow SA \mid a$ into Greibach normal form.

4. (a) Consider the grammar $G$ whose productions are $S \rightarrow aS \mid AB, \ A \rightarrow \Lambda, \ B \rightarrow \Lambda, \ D \rightarrow b.$ Construct a grammar $G_1$ without null ($\Lambda$) productions generating $L(G)^* \setminus \{\Lambda\}$. 

(Continued)
(b) Construct a push down automaton accepting 
$L = \{ a^n b^{2n} | n \geq 1 \}$ over \{a, b\}.  

[ Internal Assessment : 5 Marks ]

(Compiler Design)

MODULE – 2

[ Marks : 25 ]

Answer any two questions

1. (a) Convert \( r = (a \mid b)^* \text{ bba} \) regular expression directly to DFA using nullable, firstpos, lastpos and follow pos function.

   (b) "No left-recursive or ambiguous grammar can be LL(1)" — Justify.  

2. (a) Why LR parser is good and attractive?

   (b) Show that the following grammar

   \[ S \rightarrow SA \mid A \]
   \[ A \rightarrow a \]

   is SLR(1) but not LL(1).
(c) The shrink process of LALR may introduce reduce/reduce conflict. — Explain with example.

3 + 4 + 3

3. (a) Explain Basic Block and flow graph.

(b) Consider the three address code below:

(0) PROD = 0
(1) I = 1
(2) T₁ = 4 * I
(3) T₂ = addr(A) – 4
(4) T₃ = addr(B) – 4
(5) T₅ = T₄ [T₁]
(6) T₆ = T₃ * T₅
(7) PROD = PROD + T₆
(8) I = I + 1
(9) If i <= 20 go to (3)

(i) Find the basic block and flow graph of above sequence.

(ii) Optimize the code sequence by applying function preserving transformation and optimization techniques.
4. Write short notes on the following (any two): $5 \times 2$

(i) Handle and viable prefixes
(ii) Syntax analysis
(iii) Three address code
(iv) Basic blocks and flow graphs.

[Internal Assessment: 5 Marks]